APPENDIX G

PACKET CONSTRUCTION AND BIT ORDERING

G.1. General.

G.1.1 Scope. This appendix illustrates the construction of packets starting with the Application Layer Protocol Data Unit (PDU) and VMF Message data buffers and ending with the data link bit order of transmission and physical layer PDU. However this example excludes the S/R protocol. The focus of this example is to show correct formatting of the 188-220 subnetwork.

G.1.2 Application. This appendix is a mandatory part of this document. The bit ordering defined herein shall be utilized by all implementers.

G.2. Applicable Documents.

a. RFC 768: User Datagram Protocol

b. RFC 791: Internet Protocol -- DARPA Internet Program Protocol Specification

c. MIL-STD-2045-47001: Interoperability Standard for Connectionless Data Transfer -- Application Standard

d. Joint Interoperability of Tactical Command and Control Systems, Variable Message Format Technical Interface Design Plan (Test Edition), Reissue 2, Volume III

G.3. PDU construction. This section provides examples illustrating the construction and bit ordering of a VMF message through the Application Layer, the Transport Layer, the Network Layer, Link Layer and Physical Layer. For clarity, each layer will be discussed separately and then combined for actual transmission. The same representations will be utilized for each layer:

- the MSB (2^n bit) is represented with an italicized font and
- the LSB (2^0 bit) is shown to the RIGHT in the Value (binary) column.

This representation is carried into the other columns to identify the beginning and end of each of the fields as the bits are moved into individual octets. Note that the bit markings for MSB and LSB are on a field basis, not on an octet basis. Single bit fields are treated as LSB. In addition, since some layers (e.g. transport) are based on commercial standards, the representation from the appropriate RFC will also be included. In all cases, we will start with a figure which illustrates the interaction with upper/lower communication layers, followed by a figure showing the exchange between communication layers. There will be a table showing the construction of the PDU. This will be followed by a table showing the construction of each octet and a figure showing the serial representation of this particular PDU as it would appear at physical layer.

Each layer typically adds value and its own header to an outgoing message. This process is illustrated in Figure G-1.

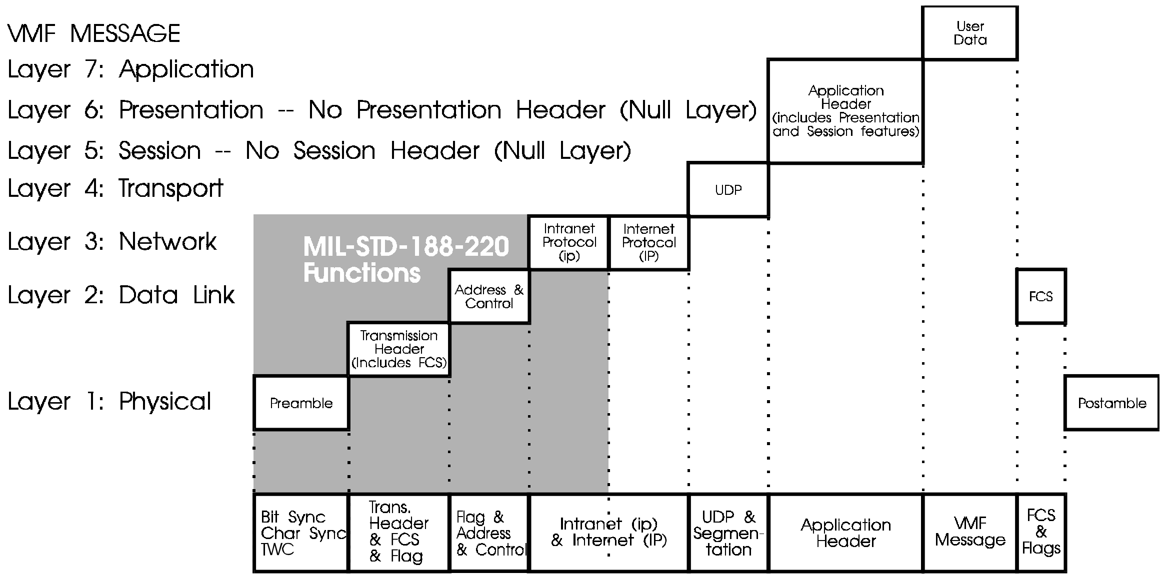

FIGURE G-1. PDU construction.

An application header is added to the VMF message at application layer. For this protocol, layers 5 and 6 are null layers, and no processing or headers are present. The Application Layer handles these functions. The transport layer adds its header. Although the standard calls out TCP, UDP and segmentation/reassembly, only UDP is illustrated in this appendix. Next, the network layer adds the IP header and the Intranet header. The message is now passed to the data link layer which adds both a header and a trailer. Finally, the physical layer adds its header resulting in the final PDU for transmission. Note that this example does not include TCP, segmentation & reassembly, or COMSEC.

G.3.1 VMF message data exchange. The relationship of the VMF Messaging Services to other communication layers is shown in Figure G-2.

A layered communication model is used in this example for consistency with the principles of the ISO OSI reference model. The model discussed here is tailored to focus attention specifically on VMF Messaging Services, and the data it produces. A user of VMF Messaging Services exchanges Message Content with its peer at another node by sending and receiving the Message Content via the VMF Messaging Services. VMF Messaging Services sends and receives the Message Content by converting the Message Content to Message Data and exchanging the Message Data with its peer at another node. The VMF Message Data is sent and received via lower communication layers. The lower communication layers send and receive the VMF Message Data transparently over a variety of communications media. Note that VMF Messaging Services would ordinarily use Application Layer services from the lower communication layers to send and receive Message Data. The Message Data would then appear in the Application Layer PDU's VMF message.

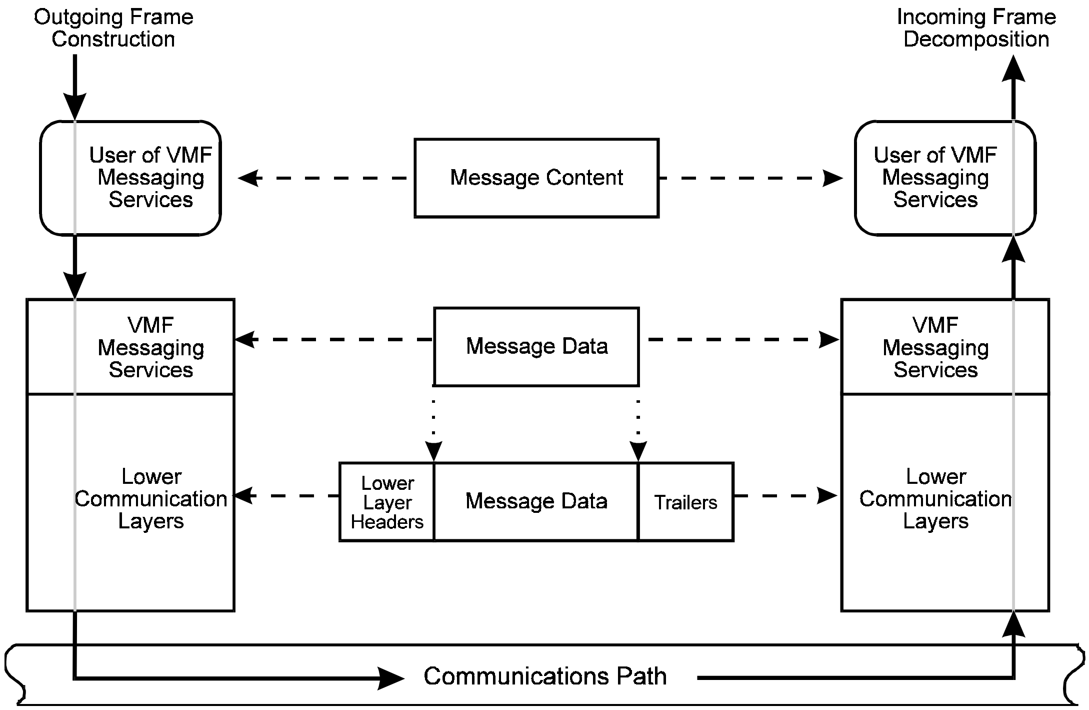

FIGURE G-2. VMF message services interaction with other communication layers.

The format of the Message Data is defined in terms of the actual data buffer or data stream used to exchange the Message Data between the VMF Messaging Services and the lower communication layers. The rationale for using the Message Data's data buffer/stream to define the format is: 1) for consistency with industry standard commercial communications hardware and software (e.g., UNIX implementations of TCP/IP), which exchange data with other software when sending or receiving as a buffer or stream of octets; 2) to provide a definition independent of the specifics of any other communication layer, consistent with the OSI ISO model principle of making communication layers independent; and 3) to avoid differences in the bit representations used to implement communications on different media. For example, on Ethernet LAN media each octet is sent least significant bit first, but on FDDI media each octet is sent most significant bit first. To achieve a universal definition of the Message Data format, its representation is defined independent of the other communication layers. The relationship of the Message Data's data buffer/stream to the VMF Messaging Services is depicted in Figure G-3. The Message Data is defined as a buffer or stream of octets. The rational for treating the Message Data as a series of octets is for consistency with the way communications data is handled by industry standard commercial communications hardware and software and for independence from platform-dependent byte ordering issues.

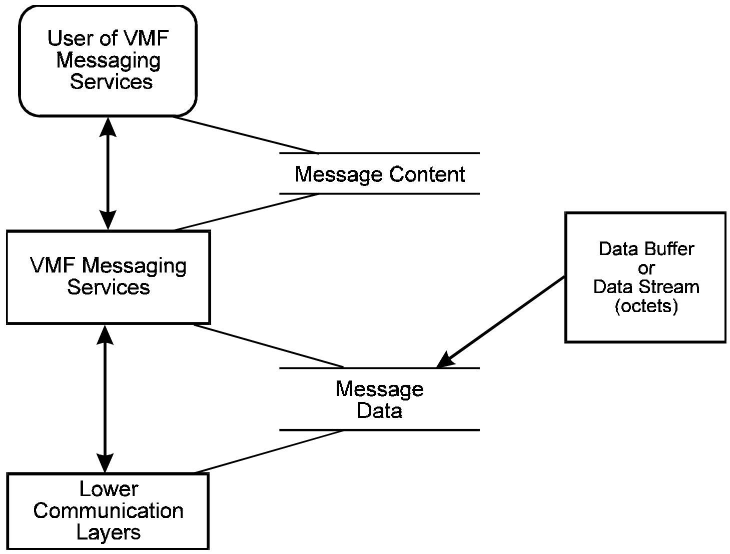

FIGURE G-3. Exchange of message data between communication layers.

G.3.1.1 Example of VMF message data construction. The construction of VMF Message Data is illustrated by the example in Table G-1. The first four columns of the table provide a description of each field in the example, the field length in bits, and the value of the field in both decimal and binary representations. The last three columns show the physical encoding of the VMF Message Data. In the fifth column, Field Fragments, the bits of each field are placed in octets. The bit(s) of each field are positioned in an octet such that the LSB of the field is positioned in the least significant unencoded bit of the octet, the next LSB of the field is placed in the next least significant unencoded bit of the octet, and repeated until all of the bits of the field have been encoded. When an octet is filled before all the bits of a field are encoded, the process is continued encoding the next octet with the remaining bits of the field. This field/octet encoding procedure is performed starting with the first field and octet, and repeated for each successive field and individual octet, in order, until the encoding is completed. When a field has groups, the field encoding procedure is performed starting with the first group, and repeated for each successive group and individual octet, in order, until the encoding of the field is completed. The Target Number field illustrates the encoding of a field with groups. Note the LSB of a field or octet is defined as the bit having the weight of 2^0 when the field or octet is represented as a numeric value. X's are used to identify bits that are not associated with the field being encoded. The sixth column, Octet Value - Binary, assembles the bits contributed by successive fields into complete octets, represented in binary. The seventh column, Octet Value – Hexadecimal, represents the octet value in hexadecimal. The last column, Octet Number, numbers the octets from first to last starting with 0.

When all fields have been encoded, any remaining unencoded bits in the last octet are filled with zeroes (zero padded). Each VMF Message is individually encoded and zero padded.

TABLE G-1. Example of VMF message data construction.

Field Name	**Length (bits)**	**Value (Dec)**	**Value (Bin)**	**Field Fragments**	**Octet Value (Bin)**	**Octet Value (Hex)**	**Octet Number**
			MSB ***2ⁿ*** **LSB** **2^0**	**MSB** **2^7** **LSB** **2^0**	**MSB** **2^7** **LSB** **2^0**		
Check Fire Type	3	0	*0*00	xxxxx*0*00			
Check Fire/Cancel Check fire command	3	1	*0*01	xx*0*01xxx			
FPI	1	1	1	x1xxxxxx			
Target Number (Group 1)	7	65 (A)	*1*000001	1xxxxxxx xx*1*00000	11*0*01*0*00	C8	0
(Group 2)	7	66 (B)	*1*000010	10xxxxxx xxx*1*0000	10*1*00000	A0	1
(Group 3)	14	1543	*0*0011000000111	111xxxxx 11000000 xxxxx*0*00	111*1*0000 11000000	F0 C0	2 3
FPI (Observer URN)	1	0	0	xxxx0xxx			
FPI (First Unit URN)	1	0	0	xxx0xxxx			
GPI DTG	1	0	0	xx0xxxxx			
FPI (launcher message)	1	0	0	x0xxxxxx			
(Zero Padding)	1	0	0	0xxxxxxx	00000*0*00	00	4

Figure G-4 illustrates the octets arranged in a serial format as they would appear at the physical layer, with LSB first.

Octet 0	**Octet 1**	**Octet 2**	**Octet 3**	**Octet 4**
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
00*0*10*0*11	00000*1*01	0000*1*111	00000011	00*0*00000

FIGURE G-4. Serial representation of PDU.

G.3.2 Application Layer Data Exchange. The relationship of the Application Layer to other communication layers is shown in Figure G-5. A layered communication model is used in this example for consistency with the principles of the ISO OSI reference model. The model discussed here is tailored to focus attention specifically on the Application Layer, and the data it produces. A user of the Application Layer exchanges a VMF message with its peer at another node by sending and receiving the VMF message via the Application Layer. The Application Layer sends and receives the VMF message transparently by producing and exchanging an Application Layer Protocol Data Unit (PDU) with its peer at another node. The Application Layer PDU consists of the Application Header concatenated with the VMF message, and is sent and received via lower communication layers. The lower communication layers send and receive the VMF message transparently over a variety of communications media.

The format of the Application Layer PDU is defined in terms of the actual data buffer or data stream used to exchange the PDU between the Application Layer and the lower communication layers. The rationale for using the PDU's data buffer/stream to define the format is 1) for consistency with industry standard commercial communications hardware and software (e.g., UNIX implementations of TCP/IP), which exchange data with other software when sending or receiving as a buffer or stream of octets; 2) to provide a definition independent of the specifics of any other communication layer, consistent with the OSI ISO model principle of making communication layers independent; and 3) to avoid differences in the bit representations used to implement communications on different media. For example, on Ethernet LAN media each octet is sent least significant bit first, but on FDDI media each octet is sent most significant bit first. To achieve a universal definition of the PDU format, its representation is defined independent of the other communication layers.

The relationship of the Application Layer PDU's data buffer/stream to the Application Layer is depicted in Figure G-6. The Application Layer PDU is defined as a buffer or stream of octets. The rational for treating the PDU as a series of octets is for consistency with the way communications data is handled by industry standard commercial communications hardware and software and for independence from platform-dependent byte ordering issues. The Application Header and the VMF message are each individually defined as a series of octets for the same reasons.

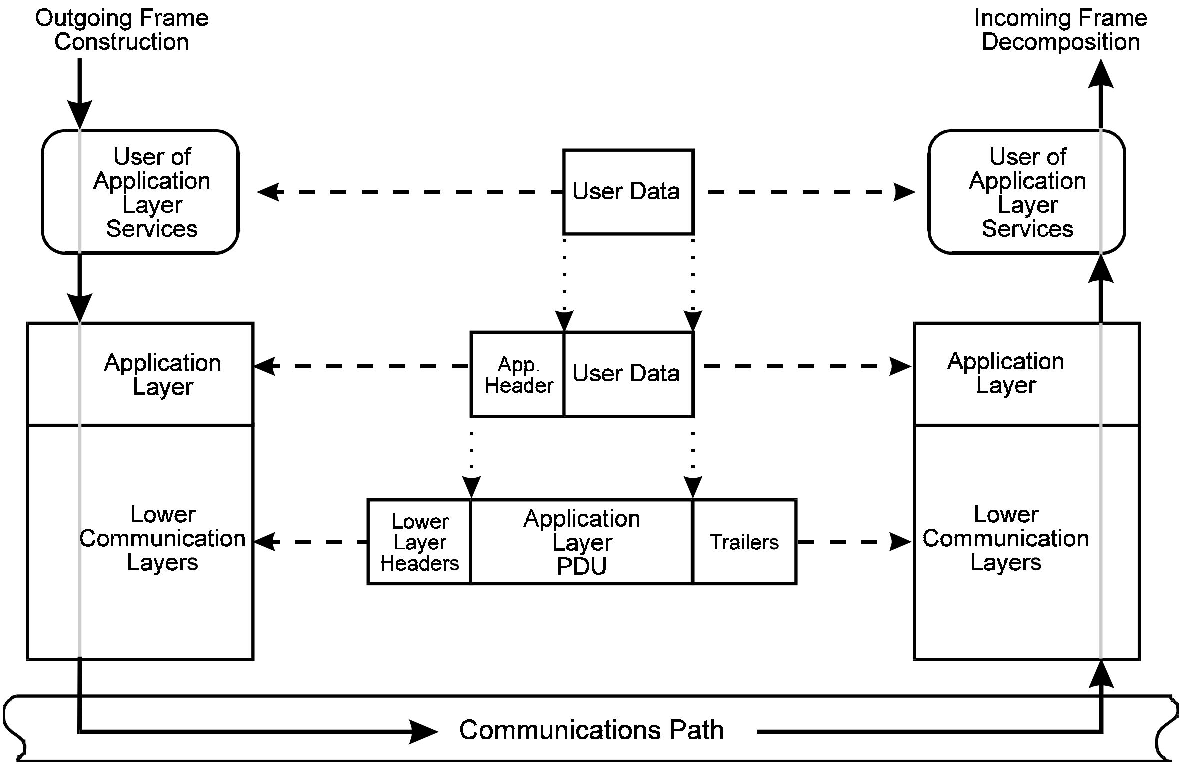

FIGURE G-5. Application layer interaction with other communication layers.

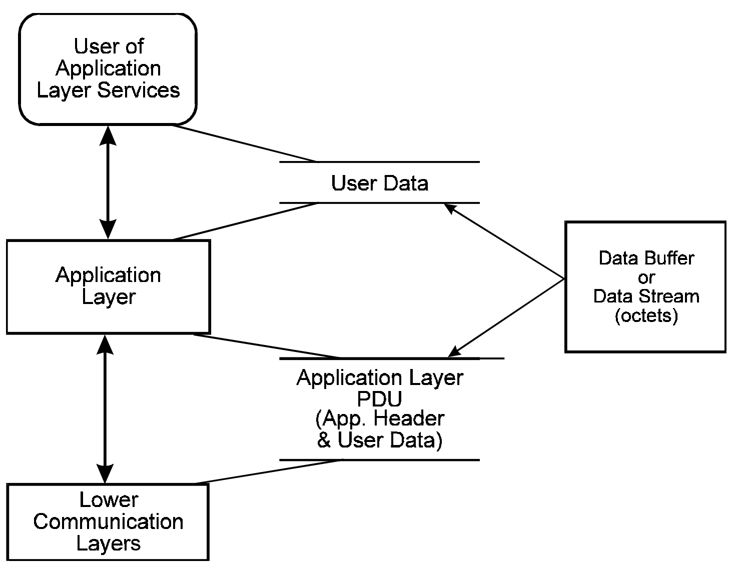

FIGURE G-6. Exchange of application layer PDU between communication layers.

G.3.2.1 Example of application layer PDU. The construction of an Application Layer PDU is illustrated by the example in Table G-2. The first four columns of the table provide a description of each field in the example, the field length in bits, and the value of the field in both decimal and binary representations. The last four columns show the physical encoding of the Application Layer PDU. In the fifth column, Field Fragments, the bits of each field are placed in octets. The bit(s) of each field are positioned in an octet such that the LSB of the field is positioned in the least significant unencoded bit of the octet, the next LSB of the field is placed in the next least significant unencoded bit of the octet, and repeated until all of the bits of the field have been encoded. When an octet is filled before all the bits of a field are encoded, the process is continued encoding the next octet with the remaining bits of the field. This field/octet encoding procedure is performed starting with the first field and octet, and repeated for each successive field and individual octet, in order, until the encoding is completed. When a field has groups, the field encoding procedure is performed starting with the first group, and repeated for each successive group and individual octet, in order, until the encoding of the field is completed. The Unit Reference Number field illustrates the encoding of a field with groups. Note the LSB of a field or octet is defined as the bit having the weight of 2^0 when the field or octet is represented as a numeric value. X's are used to identify bits that are not associated with the field being encoded. The sixth column, Octet Value - Binary, assembles the bits contributed by successive fields into complete octets, represented in binary. The seventh column, Octet Value, represents the octet value in binary that should be submitted to the Transport layer. The last column, Octet Number, numbers the octets from first to last starting with 0.

When all fields have been encoded, any remaining unencoded bits in the last octet are filled with zeroes (zero padded). The Application Header is individually encoded and zero padded. The VMF message is individually encoded and zero padded before it is passed to the Application Layer to have the Application Header added.

Any of the ASCII fields (e.g. Unit Name) in the application header can be terminated by either an end of text marker, or by using the maximum number of bits. Table G-3 shows how to format the Unit Name when the Unit Name is used as part of the originator address group. The Unit Name and Unit Reference Number are mutually exclusive inside the address group – never send both, Unit Name and Unit Reference Number, in an address group. However if the address group has a Group Repeat Indicator (GRI) each of the repeatable address groups can be different address types (e.g. Unit Name or Unit Reference Number).

The Application Header is followed by the VMF message. The VMF message is shown as a single 10-octet message to complete the Application Layer PDU.

Figure G-7 provides an illustration of the Application Header as it would appear in serial form at the lower layers.

TABLE G-2. Example construction of the application header.

Syntax	Field Description	Length (bits)	Value (Dec)	Value (Bin)	Field Fragments	Octet Value (Bin)	Octet Number
				MSB ***2^n*** **LSB** **2^0**	**MSB** **2^n** **LSB** **2^0**	**MSB** **2^n** **LSB** **2^0**	
	Version	4	1	*0*001	xxxx*0*001		
FPI	Compression Type	1	0	0	xxx0xxxx		
GPI	Presence Indicator (Originator)	1	1	1	xx1xxxxx		
FPI	Presence Indicator (URN)	1	1	1	x1xxxxxx		
	Unit Reference Number (Originator)	24	23	*0*00000000000000000010111	1xxxxxxx 00001011 00000000 x*0*000000	1110*0*001 00001011 00000000	0 1 2
FPI	Presence Indicator (Unit Name)	1	0	0	0xxxxxxx	0*0*000000	3
GPI	Presence Indicator (Recipient)	1	1	1	xxxxxxx1		
GRI	Group Repeat Indicator (Recipient)	1	0	0	xxxxxx0x		
FPI	Presence Indicator (URN)	1	1	1	xxxxx1xx		
	Unit Reference Number (Recipient URN)	24	124	*0*00000000000000001111100	11100xxx 00000011 00000000 xxxxx*0*00	11100101 00000011 00000000	4 5 6
FPI	Presence Indicator (Unit Name)	1	0	0	xxxx0xxx		
GPI	Group Presence Indicator (Information)	1	0	0	xxx0xxxx		
GRI	Group Repeat Indicator (Message)	1	0	0	xx0xxxxx		
	User Message Format	4	2	*0*010	10xxxxxx xxxxxx*0*0	10000*0*00	7
GPI	Group Presence Indicator	1	1	1	xxxxx1xx		

TABLE G-2. Example construction of the application header.

Syntax	**Field Description**	**Length (bits)**	**Value (Dec)**	**Value (Bin)**	**Field Fragments**	**Octet Value (Bin)**	**Octet Number**
				MSB 2^n **LSB** 2^0	**MSB** 2^n **LSB** 2^0	**MSB** 2^n **LSB** 2^0	
	(Message Identification)						
	Functional Area Designator	4	2	*0*010	x*0*010xxx		
	Message Number	7	1	*0*000001	1xxxxxxx xx*0*00000	1*0*0101*0*0	8
FPI	Presence Indicator (Message Subtype #)	1	0	0	x0xxxxxx		
FPI	Presence Indicator (File Name)	1	0	0	0xxxxxxx	00*0*00000	9
FPI	Presence Indicator (Message Size)	1	0	0	xxxxxxx0		
	Operation Indicator	2	0	*0*0	xxxxx*0*0x		
	Retransmit Indicator	1	0	0	xxxx0xxx		
	Message Precedence Code	3	7	*1*11	x*1*11xxxx		
	Security Classification	2	0	*0*0	0xxxxxxx xxxxxxx*0*	0*1*110*0*00	10
FPI	FPI for Control/Release Marking	1	0	0	xxxxxx0x		
GPI	GPI for Originator DTG	1	1	1	xxxxx1xx		
	Year	7	96	*1*100000	00000xxx xxxxxx*1*1	0000010*0*	11
	Month	4	7	*0*111	xx*0*111xx		
	Day	5	1	*0*0001	01xxxxxx xxxxx*0*00	01*0*111*1*1	12
	Hour	5	8	*0*1000	*0*1000xxx	*0*1000*0*00	13
	Minute	6	32	*1*00000	xx*1*00000		
	Second	6	16	*0*10000	00xxxxxx xxxx*0*100	00*1*00000	14
FPI	DTG Extension	1	0	0	xxx0xxxx		
GPI	GPI for Perishability	1	0	0	xx0xxxxx		

TABLE G-2. Example construction of the application header.

Syntax	Field Description	Length (bits)	Value (Dec)	Value (Bin)	Field Fragments	Octet Value (Bin)	Octet Number
				MSB 2^n LSB 2^0	MSB 2^n LSB 2^0	MSB 2^n LSB 2^0	
	DTG						
GPI	GPI for ACK Request Group	1	0	0	x0xxxxxx		
GPI	GPI for Response Data Group	1	0	0	0xxxxxxx	0000*0*100	15
GPI	GPI for Reference Message Data	1	0	0	xxxxxxx0		
	(Zero Padding)	7	0	*0*000000	*0*000000x	*0*0000000	16

TABLE G-3. Example of a unit name as originator.

Syntax	Field Description	Length (bits)	Value (Dec)	Value (Bin)	Field Fragments	Octet Value (Bin)	Octet Number
				MSB ***2^n*** **LSB** **2^0**	**MSB** **2^n** **LSB** **2^0**	**MSB** **2^n** **LSB** **2^0**	
	Version	4	1	0001	xxxx0001		
FPI	Compression Type	1	0	0	xxx0xxxx		
GPI	Presence Indicator (Originator)	1	1	1	xx1xxxxx		
FPI	Presence Indicator (URN)	1	0	0	x0xxxxxx		
FPI	Presence Indicator (Unit Name)	1	1	1	1xxxxxxx	10000000	0
	Unit Name (Originator)	448 Max	"UNITA"				
	"U"	7	85	1010101	x1010101		
	"N"	7	78	1001110	0xxxxxxx xx100111	01010101	1
	"I"	7	73	1001001	01xxxxxx xxx10010	01100111	2
	"T"	7	84	1010100	100xxxxx xxxx1010	10010010	3
	"A"	7	65	1000001	0001xxxx xxxxx100	00011010	4
	End of text marker (ANSI ASCII DEL)	7	127	1111111	11111xxx xxxxxx11	11111100	
GPI	Presence Indicator (Recipient)	1	1	1	xxxxx1xx		
encode rest of the message as in Figure G-3							

Octet 0	**Octet 1**	**Octet 2**		**Octet 11**	**Octet 12**	**Octet 13**	**Octet 14**	**Octet 15**	**Octet 16**
2^0 2^7	**2^0 2^7**	**2^0 2^7**		**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**
10000111	11010000	00000000		00100000	11111010	00000010	00000100	00100000	00000000

FIGURE G-7. Application header (octets).

G.3.3 Transport layer data exchange. The relationship of the Transport Layer to other communication layers is shown in Figure G-8. A user of the Transport Layer exchanges data with its peer at another node by sending and receiving the Application Layer PDU via the Transport Layer. The Transport Layer sends and receives the Application Layer PDU transparently by producing and exchanging a Transport Layer Protocol Data Unit (PDU) with its peer at another node. The Transport Layer PDU consists of the Transport Header concatenated with the Application Layer PDU, and is sent and received via lower layer communication layers. The lower communication layers send and receive the Transport PDU transparently over a variety of communications media.

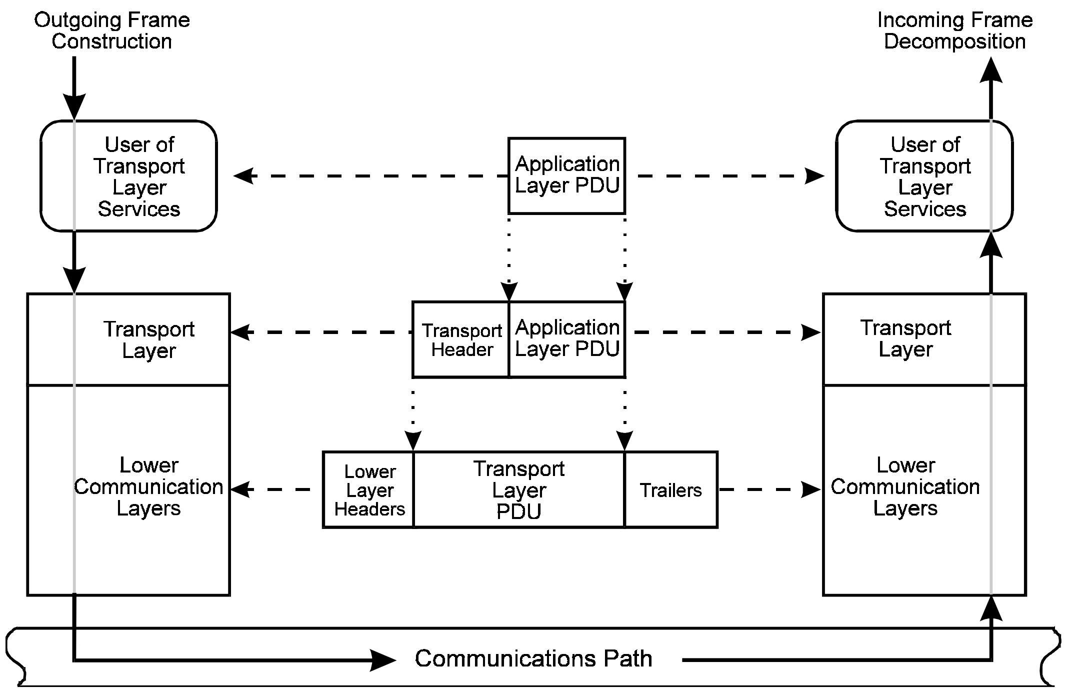

FIGURE G-8. Transport layer interaction with other communication layers.

The relationship of the Transport Layer PDU's data buffer/stream to the Application Layer is depicted in Figure G-9. The Transport Layer PDU is defined as a buffer or stream of octets consisting of the VMF message, Application Header and Transport Header.

G.3.3.1 An example of UDP header construction. UDP is described by RFC 768. The UDP header from RFC 768 consists of 8 octets as shown in Figure G-10 with the example values to be used for this appendix. Since the RFC treats bit 0 as most significant bit (MSB), Figures G-10 and G-11 show B_0 as MSB. For this example, the source has a value of 1581, destination of 1581, length of 30 and the checksum equals 3491. MIL-STD-188-220 typically treats the least significant bit as bit 0.

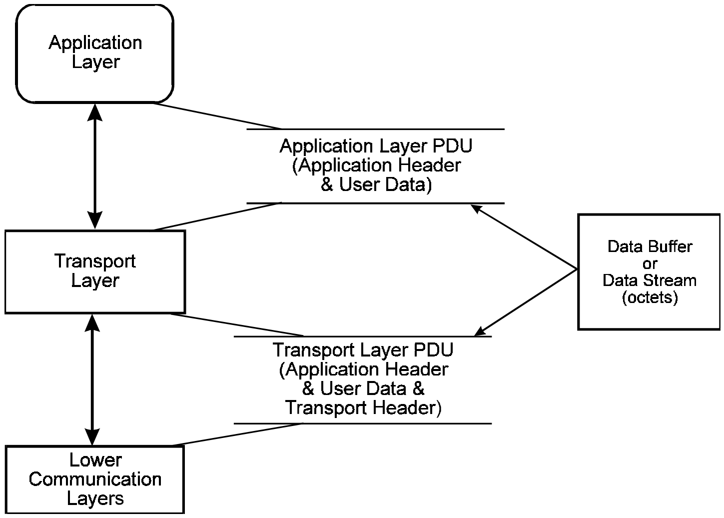

FIGURE G-9. Exchange of transport layer PDU between communication layers.

0 7	**8 15**	**16 23**	**24 31**
UDP Source (1581)		UDP Destination (1581)	
UDP Length (30)		UDP Checksum (3491)	

Bit 0 is most significant bit (MSB)

FIGURE G-10. UDP header.

Figure G-11 illustrates the eight octets comprising UDP with the binary bit patterns. Each octet is marked to show both the MSB and LSB of each octet. It demonstrates how each of the octets are arranged and passed in order to next layer.

Octet 0	**Octet 1**	**Octet 2**	**Octet 3**
B_0 B_7	B_8 B_{15}	B_{16} B_{23}	B_{24} B_{31}
2^7 2^0	2^7 2^0	2^7 2^0	2^7 2^0
00000110	00101101	00000110	00101101
UDP Source (1581)		**UDP Destination (1581)**	

Octet 4	**Octet 5**	**Octet 6**	**Octet 7**
B_0 B_7	B_8 B_{15}	B_{16} B_{23}	B_{24} B_{31}
2^7 2^0	2^7 2^0	2^7 2^0	2^7 2^0
00000000	00011110	00001101	10100011
UDP Length (30)		**UDP Checksum (3491)**	

FIGURE G-11 Octet representation of UDP header.

The construction of a Transport Layer Header is illustrated by the example in Table G-4. The first four columns of the table provide a description of each field in both decimal and binary representations. The last two columns show the physical encoding of the Transport Layer PDU. In the fifth column, Field Fragments, the bits of each field are placed in octets. The bits(s) of each field are positioned in an octet such that the LSB of the field is positioned in the least significant unencoded bit of the octet, the next LSB of the field is placed in the Next last significant unencoded bit of the octet, and repeated until all of the bits of the field have been encoded. When an octet is filled before all the bits of a field are encoded, the process is continued encoding the next octet with the remaining bits of the field. This field/octet encoding procedure is performed starting with the first field and octet, and repeated for each successive field and individual octet, in order, until the encoding is completed. The sixth column, Octet Value - Binary, assembles the bits contributed by successive fields into complete octets, represented in binary. The last column, Octet Number, numbers the octets from first to last starting with 0.

TABLE G-4. Example construction of UDP header.

Field Name	**Length (bits)**	**Value (Dec)**	**Value (Bin)**	**Field Fragments**	**Octet Value (Bin)**	**Octet Number**
			MSB 2^{15} **LSB** 2^0	**MSB** 2^7 **LSB** 2^0	**MSB** 2^7 **LSB** 2^0	
UDP Source	16	1581	*0*000011000101101	*0* 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1	*0* 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1	0 1
UDP Destination	16	1581	*0*000011000101101	*0* 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1	*0* 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1	2 3
UDP Length	16	30	*0*000000000011110	*0* 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0	*0* 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0	4 5
UDP Checksum	16	3491	*0*000000010100011	*0* 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1	*0* 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1	6 7

Table G-5 illustrates the eight octets of the Transport Header showing the binary value of the octet, the octet number (0-7) and the field represented by each octet. Note that the bit with the bold italicized font identifies the MSB (2^n) of the field, not the octet.

Figure G-12 provides a serial representation of the UDP header as it would appear at the physical layer.

TABLE G-5. Octet representation of UDP header.

Octet Value (Binary)	Octet Number	Field Name
2^7 2^0		
0 0 0 0 0 1 1 0	0	**Source**
0 0 1 0 1 1 0 1	1	**Source**
0 0 0 0 0 1 1 0	2	**Destination**
0 0 1 0 1 1 0 1	3	**Destination**
0 0 0 0 0 0 0 0	4	**Length**
0 0 0 1 1 1 1 0	5	**Length**
0 0 0 0 1 1 0 1	6	**Checksum**
1 0 1 0 0 0 1 1	7	**Checksum**

Octet 0	**Octet 1**	**Octet 2**	**Octet 3**	**Octet 4**	**Octet 5**	**Octet 6**	**Octet 7**
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
0 1 1 0 0 0 0 *0*	1 0 1 1 0 1 0 0	0 1 1 0 0 0 0 *0*	1 0 1 1 0 1 0 0	0 0 0 0 0 0 0 *0*	0 1 1 1 1 0 0 0	1 0 1 1 0 0 0 *0*	1 1 0 0 0 1 0 1

FIGURE G-12. Serial representation of UDP header.

G.3.4 Network layer data exchange. The relationship of the Network Layer to other communication layers is shown in Figure G-13. A user of the Network Layer exchanges data with its peer at another node by sending and receiving the Transport Layer PDUs via the Network Layer. The Network Layer sends and receives the Transport Layer PDUs transparently by producing and exchanging a Network Layer PDU. The Network Layer PDU consists of the Network Headers concatenated with the Transport Layer PDU, and is sent and received via lower layer communication layers. The lower communication layers send and receive the Network Layer PDU transparently over a variety of communications media.

The relationship of the Network Layer PDU's data buffer/stream to the Transport Layer is depicted in Figure G-14. The Network Layer PDU is defined as a buffer or stream of octets consisting of the VMF message, Application Header, Transport Header and Network Headers. There are two Network Headers in the Network Layer PDU when using MIL-STD-188-220.

The Internet Protocol (IP) is described by RFC 791. The IP header from RFC 791 is shown in Figure G-15 with the example values to be used for this appendix.

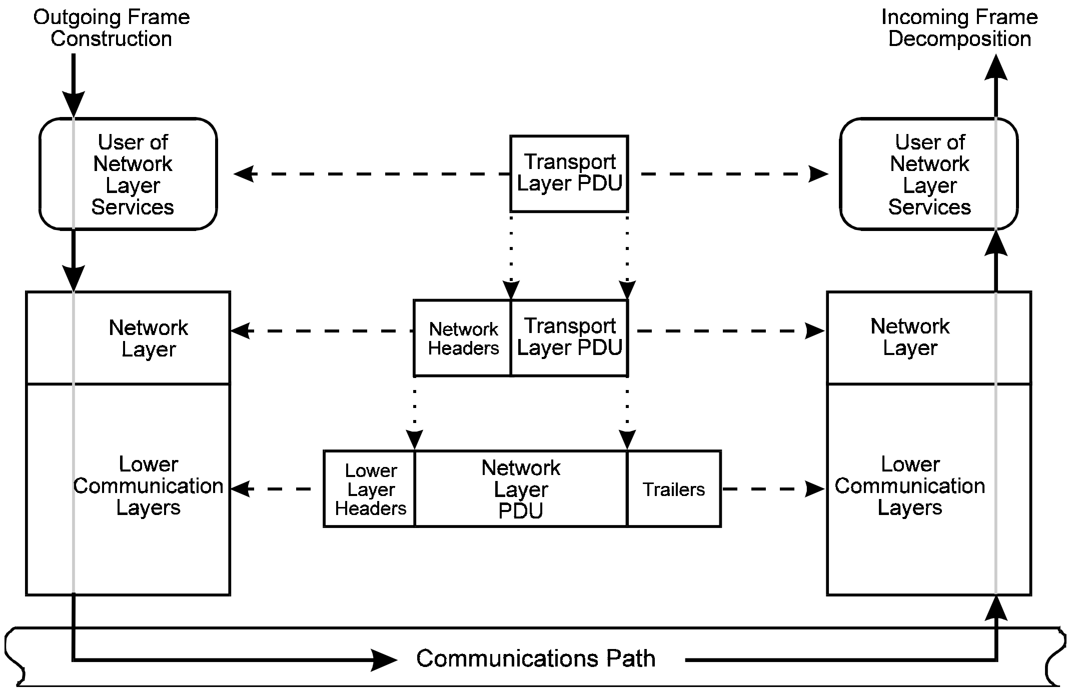

FIGURE G-13. Network layer interaction with other communication layers.

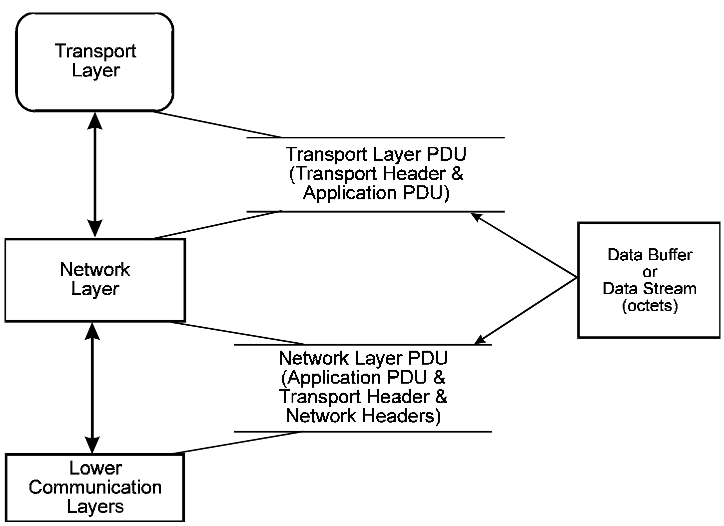

FIGURE G-14. Exchange of network layer PDU between communication layers.

<table>
<tr><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">2</td><td colspan="2">3</td></tr>
<tr><td colspan="2">0 1 2 3 4 5 6 7</td><td colspan="2">8 9 0 1 2 3 4 5</td><td colspan="2">6 7 8 9 0 1 2 3</td><td colspan="2">4 5 6 7 8 9 0 1</td></tr>
<tr><td>Ver (4)</td><td>IHL (5)</td><td colspan="2">Type of Service (0)</td><td colspan="4">Total Length (50)</td></tr>
<tr><td colspan="4">Identification (1)</td><td>Flag (0)</td><td colspan="3">Fragment Offset (0)</td></tr>
<tr><td colspan="2">Time to Live (50)</td><td colspan="2">Protocol (17)</td><td colspan="4">Header Checksum (4093)</td></tr>
<tr><td colspan="8">Source Address (192.31.124.65)</td></tr>
<tr><td colspan="8">Destination Address (192.31.124.61)</td></tr>
</table>

FIGURE G-15. IP header.

G.3.4.1 Example of internet layer header. The construction of an Internet Layer Header is illustrated by the example in Table G-6. The first four columns of the table provide a description of each field in the example, the field length in bits, and the value of the field in both decimal and binary representations. The last three columns show the physical encoding of the Internet Layer Header. In the fifth column, Field Fragments, the bits of each field are placed in octets. The bit(s) of each field are positioned in an octet such that the LSB of the field is positioned in the least significant unencoded bit of the octet, the next LSB of the field is placed in the next least significant unencoded bit of the octet, and repeated until all of the bits of the field have been encoded. When an octet is filled before all the bits of a field are encoded, the process is continued encoding the next octet with the remaining bits of the field. This field/octet encoding procedure is performed starting with the first field and octet, and repeated for each successive field and individual octet, in order, until the encoding is completed. X's are used to identify bits that are not associated with the field being encoded. The sixth column, Octet Value - Binary, assembles the bits contributed by successive fields into complete octets, represented in binary. The last column, Octet Number, numbers the octets from first to last starting with 0.

Figure G-16 illustrates the Internet Header demonstrating the relationship between the individual bits (B^0 - B^7), the bit weighting (2^7 - 2^0), the individual fields and the example bit patterns associated with each field.

TABLE G-6. Example construction of IP header.

Field Name	**Length (bits)**	**Value (Dec)**	**Value (Bin)** 2^n 2^0	**Field Fragments** 2^n 2^0	**Octet Value (Bin)** 2^n 2^0	**Octet Number**
Version	4	4	*0*100	*0*100xxxx		
Internet Header Length	4	5	*0*101	xxxx*0*101	*0*100*0*101	0
Type of Service	8	0	*0*0000000	*0*0000000	*0*0000000	1
Length	16	50	*0*000000000110010	*0*0000000 00110010	*0*0000000 00110010	2 3
Identification	16	111	*0*000000000000001	*0*0000000 00000001	*0*0000000 00000001	4 5
Flags	3	0	*0*00	*0*00xxxxx		
Fragmentation Offset	13	0	*0*000000000000	xxx*0*0000 00000000	*0*00*0*0000 00000000	6 7
Time to Live	8	50	*0*0110010	*0*0110010	*0*0110010	8
Protocol	8	17	*0*0010001	*0*0010001	*0*0010001	9
Header Checksum	16	4093	*0*000111111111101	*0*0001111 11111101	*0*0001111 11111101	10 11
Source Address	32	192.31.124.65	*1*100000000011111 0111110001000001	*1*1000000 00011111 01111100 01000001	*1*1000000 00011111 01111100 01000001	12 13 14 15
Destination Address	32	192.31.124.61	*1*100000000011111 0111110000111101	*1*1000000 00011111 01111100 00111101	*1*1000000 00011111 01111100 00111101	16 17 18 19

<table>
<tr><th colspan="2">Octet 0</th><th>Octet 1</th><th>Octet 2</th><th>Octet 3</th></tr>
<tr><td colspan="2">B_0 B_7</td><td>B_0 B_7</td><td>B_0 B_7</td><td>B_0 B_7</td></tr>
<tr><td colspan="2">2^7 2^0</td><td>2^7 2^0</td><td>2^7 2^0</td><td>2^7 2^0</td></tr>
<tr><td colspan="2">0 1 0 0 0 1 0 1</td><td>0 0 0 0 0 0 0 0</td><td>0 0 0 0 0 0 0 0</td><td>0 0 1 1 0 0 1 0</td></tr>
<tr><td>Ver (4)</td><td>IHL (5)</td><td>Type of Service (0)</td><td colspan="2">Total Length (50)</td></tr>
</table>

<table>
<tr><th>Octet 4</th><th>Octet 5</th><th colspan="2">Octet 6</th><th>Octet 7</th></tr>
<tr><td>B_0 B_7</td><td>B_0 B_7</td><td colspan="2">B_0 B_7</td><td>B_0 B_7</td></tr>
<tr><td>2^7 2^0</td><td>2^7 2^0</td><td colspan="2">2^7 2^0</td><td>2^7 2^0</td></tr>
<tr><td>0 0 0 0 0 0 0 0</td><td>0 0 0 0 0 0 1</td><td>0 0 0</td><td>0 0 0 0 0</td><td>0 0 0 0 0 0 0 0</td></tr>
<tr><td colspan="2">Identification (1)</td><td>Flag (0)</td><td colspan="2">Fragment Offset (0)</td></tr>
</table>

<table>
<tr><th>Octet 8</th><th>Octet 9</th><th>Octet 10</th><th>Octet 11</th></tr>
<tr><td>B_0 B_7</td><td>B_0 B_7</td><td>B_0 B_7</td><td>B_0 B_7</td></tr>
<tr><td>2^7 2^0</td><td>2^7 2^0</td><td>2^7 2^0</td><td>2^7 2^0</td></tr>
<tr><td>0 0 1 1 0 0 1 0</td><td>0 0 0 1 0 0 0 1</td><td>0 0 0 0 1 1 1 1</td><td>1 1 1 1 1 1 0 1</td></tr>
<tr><td>Time (50)</td><td>Protocol (17)</td><td colspan="2">Header Checksum (4093)</td></tr>
</table>

<table>
<tr><th>Octet 12</th><th>Octet 13</th><th>Octet 14</th><th>Octet 15</th></tr>
<tr><td>B_0 B_7</td><td>B_0 B_7</td><td>B_0 B_7</td><td>B_0 B_7</td></tr>
<tr><td>2^7 2^0</td><td>2^7 2^0</td><td>2^7 2^0</td><td>2^7 2^0</td></tr>
<tr><td>1 1 0 0 0 0 0 0</td><td>0 0 0 1 1 1 1 1</td><td>0 1 1 1 1 1 0 0</td><td>0 1 0 0 0 0 0 1</td></tr>
<tr><td colspan="4">Source Address (192.31.124.65)</td></tr>
</table>

<table>
<tr><th>Octet 16</th><th>Octet 17</th><th>Octet 18</th><th>Octet 19</th></tr>
<tr><td>B_0 B_7</td><td>B_0 B_7</td><td>B_0 B_7</td><td>B_0 B_7</td></tr>
<tr><td>2^7 2^0</td><td>2^7 2^0</td><td>2^7 2^0</td><td>2^7 2^0</td></tr>
<tr><td>1 1 0 0 0 0 0 0</td><td>0 0 0 1 1 1 1 1</td><td>0 1 1 1 1 1 0 0</td><td>0 0 1 1 1 1 0 1</td></tr>
<tr><td colspan="4">Destination Address (192.31.124.61)</td></tr>
</table>

FIGURE G-16. Octet representation of IP header.

G.3.4.2 Example of intranet layer header. The construction of an Intranet Layer Header is illustrated by the example in Table G-7. The first four columns of the table provide a description of each field in the example, the field length in bits, and the value of the field in both decimal and binary representations. The last three columns show the physical encoding of the Intranet Layer Header. In the fifth column, Field Fragments, the bits of each field are placed in octets. The bit(s) of each field are positioned in an octet such that the LSB of the field is positioned in the least significant unencoded bit of the octet, the next LSB of the field is placed in the next least significant unencoded bit of the octet, and repeated until all of the bits of the field have been encoded. When an octet is filled before all the bits of a field are encoded, the process is continued encoding the next octet with the remaining bits of the field. This field/octet encoding procedure is performed starting with the first field and octet, and repeated for each successive field and individual octet, in order, until the encoding is completed. X's are used to identify bits that are not associated with the field being encoded. The sixth column, Octet Value - Binary, assembles the bits contributed by successive fields into complete octets, represented in binary. The last column, Octet Number, numbers the octets from first to last starting with 0. This example only illustrates the Intranet Header fields that must be transmitted as a minimum.

TABLE G-7. Example construction of Intranet header (minimum).

Field Name	**Length (bits)**	**Value (Dec)**	**Value & Byte Representation (Bin)** 2^n 2^0	**Field Fragments** 2^7 2^0	**Octet Value (Bin)** 2^7 2^0	**Octet Number**
Version Number	4	0	*0*000	xxxx*0*000		
Message Type	4	4	*0*100	*0*100xxxx	*0*100*0*000	0
Intranet Header Length	8	3	*0*0000011	*0*0000011	*0*0000011	1
Type of Service	8	0	*0*0000000	*0*0000000	*0*0000000	2

The Intranet layer is defined in MIL-STD-188-220 and is shown in Figure G-17 with the example values used in this appendix.

Octet 0		**Octet 1**	**Octet 2**
2^0	2^7	2^0 2^7	2^0 2^7
0 0 0 *0*	0 0 1 *0*	1 1 0 0 0 0 0 *0*	0 0 0 0 0 0 0 *0*
Version (0)	**Message Type (4)**	**Intranet Header Length (3)**	**Type of Service (0)**

FIGURE G-17. Intranet header.

Figure G-18 provides a serial representation of the Network Layer PDU as it would appear at the physical layer.

Intranet header			**IP header**				
Octet 0	**Octet 1**	**Octet 2**	**Octet 3**	**Octet 4**	**Octet 5**	**Octet 6**	**Octet 7**
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
000*00*01*0*	1100000*0*	0000000*0*	101*00*01*0*	0000000*0*	0000000*0*	01001100	0000000*0*

IP header (continued)							
Octet 8	**Octet 9**	**Octet 10**	**Octet 11**	**Octet 12**	**Octet 13**	**Octet 14**	**Octet 15**
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
10000000	0000000*0*	00000000	0100110*0*	1000100*0*	1111000*0*	10111111	0000001*1*

IP header (continued)						**IP header (end)**
Octet 16	**Octet 17**	**Octet 18**	**Octet 19**	**Octet 20**	**Octet 21**	**Octet 22**
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
11111000	00111110	10000010	0000001*1*	11111000	00111110	10111100

FIGURE G-18. Serial representation of network layer PDU.

G.3.6 Data link layer data exchange. The relationship of the Data Link Layer to other communication layers is shown in Figure G-19. A user of the Data Link Layer exchanges the Network Layer PDU with its peer at another node by sending and receiving the Network PDU via the Data Link Layer. The Data Link Layer sends and receives the VMF message transparently by producing and exchanging a Data Link Layer PDU with its peer at another node. The Data Link Layer PDU consists of the Transmission Header, and Data Link Frame Header, Network PDU, and the Data Link Frame Trailer, and is sent and received via the Physical layer. The Physical layer sends and receives the VMF message transparently over a variety of communications media.

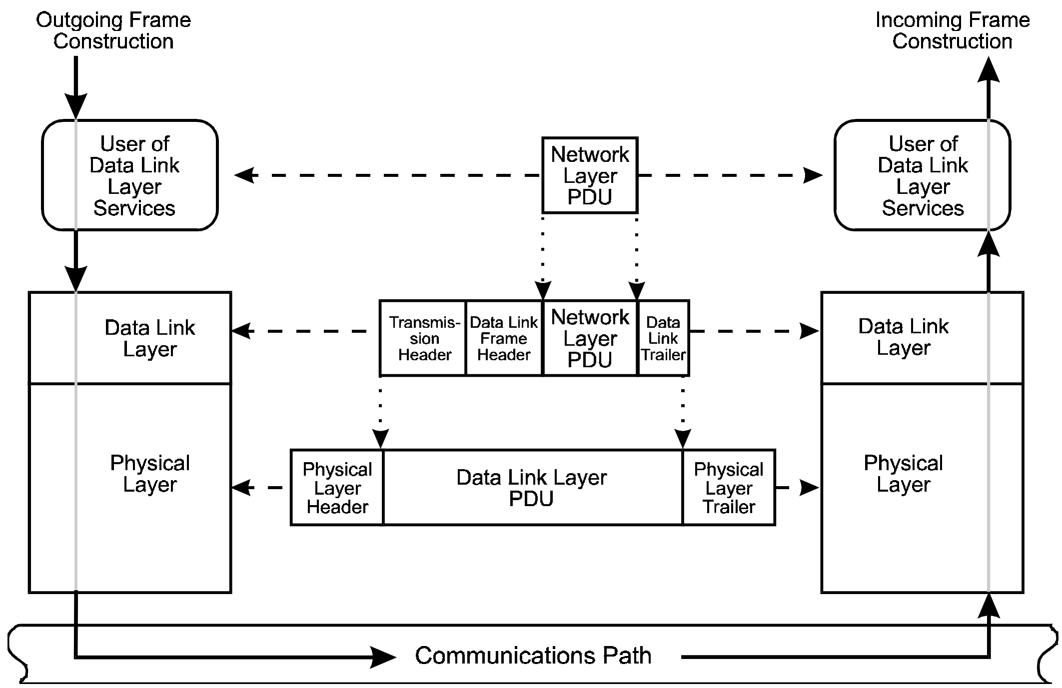

Figure G-19. Data link layer interaction with other communication layers.

The format of the Data Link Layer PDU is defined in terms of the actual data buffer or data stream used to exchange the PDU between the Network Layer and the Physical Layer. The relationship of the Data Link Layer PDU's data buffer/stream to the Intranet Layer is depicted in Figure G-20. The Data Link Layer PDU is defined as a buffer or stream of octets consisting of the Transmission Header, Data Link Frame Header, Network PDU and Data Link Layer trailer.

G.3.6.1 Example of data link layer PDU. The Data Link Layer PDU consists of the Transmission Header, Data Link Frame Header, followed by the information field and Data Link Frame Trailer as shown in Figure G-21. The information field consists of the Network PDU described previously (VMF message, Application Header, Transport Header, IP Header and Intranet Header).

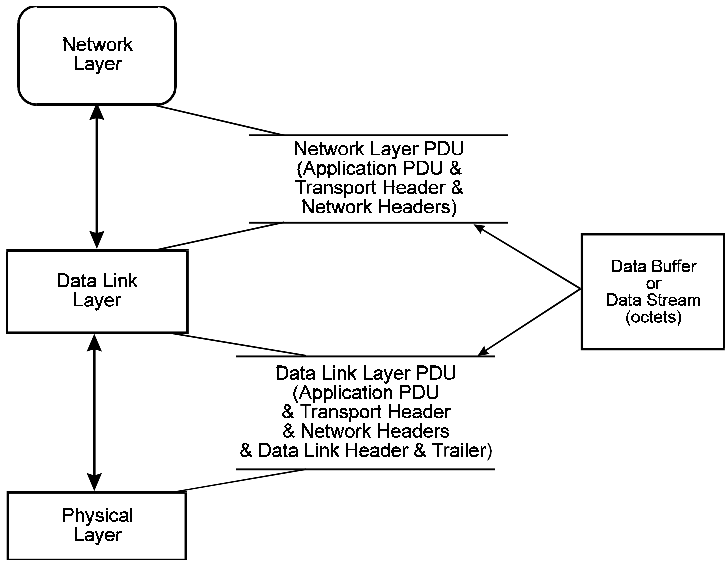

FIGURE G-20. Exchange of data link layer PDU between communication layers.

Trans-mission Header	**Data Link Frame Header**	**Information Field**	**Data Link Frame Trailer**

FIGURE G-21. Data link layer PDU.

Table G-8 illustrates the Data Link Frame Header, and Table G-9 illustrates the Data Link Frame Trailer. The first four columns of the tables provide a description of each field in the example, the field length in bits, and the value of the field in both decimal and binary representations. The last three columns show the physical encoding of the Data Link Frame. In the fifth column, Field Fragments, the bits of each field are placed in octets. The bit(s) of each field are positioned in an octet such that the LSB of the field is positioned in the least significant unencoded bit of the octet, the next LSB of the field is placed in the next least significant unencoded bit of the octet, and repeated until all of the bits of the field have been encoded. When an octet is filled before all the bits of a field are encoded, the process is continued encoding the next octet with the remaining bits of the field. This field/octet encoding procedure is performed starting with the first field and octet, and repeated for each successive field and individual octet, in order, until the encoding is completed. The sixth column, Octet Value - Binary, assembles the bits contributed by successive fields into complete octets, represented in binary. The last column, Octet Number, numbers the octets from first to last starting with 0.

TABLE G-8. Example construction of data link frame header.

Field Name	**Length (bits)**	**Value (Dec)**	**Value (Bin)**	**Field Fragments**	**Octet Value (Bin)**	**Octet Number**
			2^n 2^0	2^7 2^0	2^7 2^0	
Flag	8	126	*0*1111110	*0*1111110	*0*1111110	0
Command/Response Bit	1	0	0	xxxxxxxx0		
Source Address	7	7	*0*000111	*0*000111x	*0*0001110	1
Extension Bit	1	1	1	xxxxxxx1		
Destination Address	7	4	*0*000100	*0*000100x	*0*0001001	2
Control Field	8	19	*0*0010011	*0*0010011	*0*0010011	3

TABLE G-9. Example construction of data link frame trailer.

Frame Check Sequence (transmitted MSB first)	32	162159487	*0*000100110101010 0101110101111111	*0*0001001 10101010 01011101 01111111	*0*0001001 10101010 01011101 01111111	0 1 2 3
Flag	8	126	*0*1111110	*0*1111110	*0*1111110	4

Table G-10 illustrates the octets comprising the Data Link Frame showing the actual bit patterns from the previous examples for each layer, the octet number based on each individual layer, and the octet number based on entire Data Link Frame. This data is shown in serial representation as it would be transmitted in Figure G-22.

TABLE G-10. Octets comprising data link frame.

2^7 2^0	Nomenclature	Octet Number (Individual Layer)	Octet Number (Entire Transaction)
*0*1111110	**Flag**	0	0
*0*0001110	**Source Address**	1	1
*0*0001001	**Destination Address**	2	2
*0*0010011	**Control Field**	3	3
*0*100*0*000		0	4
*0*0000011	***INTRANET HEADER***	1	5
*0*0000000		2	6
*0*100*0*101		0	7
*0*0000000		1	8
*0*0000000		2	9
00110010		3	10
*0*0000000		4	11
00000001		5	12
*0*00*0*0000		6	13
•	***IP HEADER***	•	•
•		•	•
01111100		18	25
00111101		19	26
*0*0000110		0	27
00101101		1	28
*0*0000110		2	29
•	***UDP HEADER***	•	•
•		•	•
10100011		7	34

TABLE G-10. Octets comprising data link frame.

2^7 2^0	**Nomenclature**	**Octet Number (Individual Layer)**	**Octet Number (Entire Transaction)**
1110*0*001		0	35
00001011		1	36
•		•	•
•		•	•
0*1*110000		10	45
0000010*0*		11	46
01*0*111*1*1	***APPLICATION HEADER***	12	47
01000*0*00		13	48
•		•	•
•		•	•
0000*0*100		15	50
*0*0000000		16	51
11*0*01*0*00		0	52
10*1*00000		1	53
111*1*0000		2	54
11000000		3	55
•	***CHECKFIRE MESSAGE***	•	•
•		•	•
00000*0*00		4	56
*0*0001001	**Note: FCS transmitted MSB First**	0	57
10101010	**FCS**	1	58
01011101		2	59
01111111		3	60
*0*1111110	**Flag**	0	61

DATA LINK FRAME HEADER				INTRANET HEADER				IP	
0	1	2	3	4		5	6	7	
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7		2^0 2^7	2^0 2^7	2^0 2^7	
FLAG	**SRC**	**DST**	**CNTL**	**V**	**T**	**LEN**	**TOS**	**L**	**V**
01111110	01110000	10010000	11001000	0000	0010	11000000	00000000	1010	0010

IP (cont.)							
8	9	10	11	12	13		14
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7		2^0 2^7
TOS	**Total Length**		**Identification**		**Offset**	**Flag**	**Offset**
00000000	00000000	01001100	00000000	10000000	00000	000	00000000

IP (cont.)		UDP			
25	26	27	28	29	30
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
DESTINATION		**SOURCE**		**DESTINATION**	
00111110	10111100	01100000	10110100	01100000	10110100

APP. HEADER		
34	35	36
2^0 2^7	2^0 2^7	2^0 2^7
CHKSM	**GPI-FPI-ORIG**	
11000101	10000111	11010000

APP. HEADER			
45	46	47	48
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
Message Size- etc.			**GPI-YR**
00001110	00100000	11111010	00000010

APP. HEADER		VMF MESSAGE			
50	51	52	53	54	55
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
Min/Sec-etc.		**CF-etc.**	**GROUP-etc**		
00100000	00000000	00010011	00000101	00001111	00000011

LINK FRAME TRAILER					
56	57	58	59	60	61
2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7	2^0 2^7
Pad	**FCS**				**FLAG**
00000000	00001001	10101010	01011101	01111111	01111110

FIGURE G-22. Serial representation of data link layer PDU.

G.3.6.1.1 Zero bit insert/v36 scramble/FEC/TDC of the data link frame. The Data Link Frame must be zero inserted to prevent any part of the data accidentally being interpreted as a Frame Flag. Also in our example scrambling, FEC and TDC are being used. Figure G-23 shows some of the example data before applying zero-bit insertion, scrambling, FEC or TDC. After zero-bit insertion, scrambling, FEC and TDC, the fields are not easy to identify; therefore field names are not shown.

1 word		**2 word**		**3 word**	
2^0 2^7	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**
0x7e70		0x90c8		0x02c0	

30 word		**31 word**		**32 word**	
2^0 2^7	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**	**2^0 2^7**
0x09aa		0x5d7f		0x7e00	

FIGURE G-23. Data before zero bit insertion in transmission order.

The following is a Hex dump of the data link frame in the different stages: (a) zero-bit inserted, (b) scrambled, (c) FEC, and (d) TDC:

Note: In the following dumps the 16 bit values are in transmission order. The TWC in the physical layer is defined in words and fields are no longer easily distinguishable.

a. Data after zero bit insertion (505 bits plus 7 padding bits)
0x7e70 0x90c8 0x02c0 0x00a2 0x0000 0x4c00 0x8000 0x004c 0x88f0 0xbe81 0xf60f
0x9040 0x7d83 0xe5e3 0x05a3 0x05a0 0x03c5 0x862c 0x3e40 0x0002 0x9f00 0x0002
0x5200 0x1c41 0xf202 0x0420 0x0013 0x050f 0x0300 0x09aa 0x5d7d 0xbf00

b. Data after V.36 scrambling (512 bits)
0x8f80 0x872a 0xa161 0x7a0a 0xbfaa 0x524c 0x50c3 0x50aa 0x024c 0x6cc2 0x9ca9
0x6b17 0xe9f3 0x0403 0xbda9 0xfe4c 0xfc54 0x3014 0x02e2 0xe3a7 0xb9fa 0xdf90
0x0006 0x2754 0xf1bf 0x5f20 0x0b70 0xe695 0x59a2 0xfc47 0x616b 0x5d41

c. Data after FEC(Golay 24,12) (data size in bits: 0x0408 plus 8 padding bits)
Golay (24,12) is derived from Golay (23,12): See paragraph F 4.1 for details.
0x8f8a 0x5a08 0x7898 0x2aae 0x8616 0x140a 0x7a0b 0xf0ab 0xf3e8 0xaa54 0x7624
0xc5a0 0x50c6 0xde35 0x0622 0xaa06 0x0a24 0xc5a0 0x6cc0 0x4029 0xc884 0xa960
0x08b1 0x7c8c 0xe9f3 0x1e30 0x424a 0x03b9 0xb8da 0x9dc0 0xfe40 0x8acf 0xc6f6
0x543d 0xc201 0x49f0 0x02e8 0xa22e 0x3632 0xa7b7 0x3c9f 0xa4d0 0xdf93 0x3e00
0x0000 0x0622 0x6a75 0x4a8e 0xf1bd 0xe6f5 0xfae8 0x200f 0x68b7 0x0c9a 0xe69b
0x5e55 0x9a5c 0xa2f3 0x54c4 0x7c94 0x6169 0x9cb5 0xd5ec 0x4105 0x5c00

d. Data after TDC(16,24) (data size in bits: 0x0480)
0x8623 0x0888 0x2f7f 0x18c1 0xee2e 0x9158 0xbe20 0x8447 0xa59c 0x479f 0x6403
0x5601 0xe805 0x33f1 0xace0 0x0d10 0x6d95 0x8e88 0x0f50 0xca80 0xd4a3 0x2285
0xb2e0 0x0000 0x9c38 0x9e09 0xc861 0x5a19 0x9c58 0x0e7b 0x3cfa 0xa539 0xb4b8
0xcd81 0xa2f2 0xb268 0x3381 0x1670 0xc46b 0xb328 0x3f91 0x5712 0x25ea 0xa578
0xe82b 0x8429 0xcecb 0x0000 0xdb40 0xcda0 0xfac0 0xd440 0x0000 0x5d40 0x1a00
0xd4e0 0xce40 0x43c0 0xc380 0xcf40 0xfd80 0xb160 0x6e00 0xaae0 0xd1c0 0xee60
0xe040 0x1fa0 0x7ce0 0x8fe0 0x9800 0x0000

G.3.6.1.2 Construction of the transmission header. The Transmission Header precedes the data link frame and formatted as defined in Table G-11.

G.3.6.1.3 Zero bit insert/v36 scramble/FEC of the transmission header. The Transmission Header must be zero inserted to prevent any part of the data accidentally being interpreted as a Frame Flag. After zero-bit insertion, the fields are not easy to identify; therefore field names are not shown. The following is a Hex dump of the Transmission Header of zero-bit inserted:

Transmission Header after zero bit insertion (Size In Bits 0x0040)
0x7ee0 0x001c 0x2119 0x707e

G.3.6.1.4 Completed data link layer PDU to be passed to the physical layer. The data link layer passes the Data Link Layer PDU to the physical layer. The elements of a Data Link Layer PDU include one transmission header and one or more PDUs. The following complete data link PDU (consisting of transmission header and data link frame) will be passed to the physical layer:

Complete Data Link Layer PDU

a. Transmission Header:
0x7ee0 0x001c 0x2119 0x707e

b. Data Link Layer Frame (72 16 bit words):
0x8623 0x0888 0x2f7f 0x18c1 0xee2e 0x9158 0xbe20 0x8447 0xa59c 0x479f 0x6403
0x5601 0xe805 0x33f1 0xace0 0x0d10 0x6d95 0x8e88 0x0f50 0xca80 0xd4a3 0x2285
0xb2e0 0x0000 0x9c38 0x9e09 0xc861 0x5a19 0x9c58 0x0e7b 0x3cfa 0xa539 0xb4b8
0xcd81 0xa2f2 0xb268 0x3381 0x1670 0xc46b 0xb328 0x3f91 0x5712 0x25ea 0xa578
0xe82b 0x8429 0xcecb 0x0000 0xdb40 0xcda0 0xfac0 0xd440 0x0000 0x5d40 0x1a00
0xd4e0 0xce40 0x43c0 0xc380 0xcf40 0xfd80 0xb160 0x6e00 0xaae0 0xd1c0 0xee60
0xe040 0x1fa0 0x7ce0 0x8fe0 0x9800 0x0000

G.3.7 Physical layer data exchange. The relationship of the Physical Layer to other communication layers is shown in Figure G-24. A user of the Physical Layer exchanges the Data Link Layer PDU with its peer at another node by sending and receiving the Data Link PDU via the Physical Layer.

TABLE G-11. Example construction of data link transmission header.

Field Name	**Length (bits)**	**Value (Dec)**	**Value (Bin)** 2^n … 2^0	**Field Fragments** 2^7 … 2^0	**Octet Value (Bin)** 2^7 … 2^0	**Octet Number**
Flag	8	126	*0*1111110	*0*1111110	*0*1111110	0
FEC	1	0	1	xxxxxxx1		
TDC	1	0	1	xxxxxx1x		
Scramble	1	0	1	xxxxx1xx		
Topology Update Id	3	0	*0*00	xx*0*00xxx		
Transmit Queue	10	0	*0*000000000	00xxxxxx	00*0*00111	1
				*0*0000000	*0*0000000	2
FCS	32	471931248	*0*001110000100001	*0*0011100	*0*0011100	3
			0001100101110000	00100001	00100001	4
				00011001	00011001	5
				01110000	01110000	6
Flag	8	126	*0*1111110	*0*1111110	*0*1111110	7

1	**2**	**3**	**85**	**86**	**87**
0x64f2	0xf296	0x905e	0xe098	0x0000	0x0000
0110010011110010	1111001010010110	1001000001011110	1110000010011000	0000000000000000	0000000000000000

FIGURE G-24. Serial representation of physical layer transmission unit.

G.3.7.1 Physical layer processing example. The Physical layer encodes data submitted by the data link layer in a format to meet the physical media's requirements. This example does not address the electrical or mechanical functions normally associated with the physical layer protocols. At the physical layer the transmission header is extracted and the TWC is calculated, the Transmission header is FEC & TDC encoded. Note the other physical layer functions (COMSEC, DMTD, etc) are not shown in this example.

TWC	*Transmission Header*	*Data Link Frame*

G.3.7.1.1 Transmit word count (TWC). TWC is calculated across the data link frame plus the size of the encoded Transmission Header & TWC size (encoded Transmission Header & TWC [10.5 16 bit words]). Therefore this Physical layer PDU's TWC would be calculated as follows:

TWC = encoded data link frame + encoded Transmission Header and TWC
TWC = 72 words + 10.5 words (rounded up to nearest word)
TWC = 83 words

TWC (83)	*Transmission Header*	*Data Link Frame*

Transmission header including TWC (size in bits: 0x004C)
0xca07 0xee00 0x01c2 0x1197 0x07e0

G.3.7.1.2 FEC & TDC of transmission header. The Transmission Header must have FEC & TDC encoding applied. Below is the Transmission Header in the different stages of FEC & TDC:

a. Transmission header/with TWC after FEC (Golay 24,12) (size in bits: 0x00a8)
Golay (24,12) is derived form Golay (23,12): See paragraph F 4.1 for details.
0xca0f 0x587e 0xe806 0x0000 0x001c 0x20c8 0x1191 0xfe70 0x75a4 0xe005 0x2600

b. Transmission header/with TWC after TDC (7,24) (size in bits: 0x00a8)
0x838d 0x1aed 0x0a30 0x0448 0x8950 0x6c10 0xe047 0x1d30 0x3c49 0x89d2 0x8000

G.3.7.1.3 The Physical layer PDU. Complete message including 64-bit frame synchronization, TWC, transmission header, and data link frame. (size in bits: 0x0568):

0x64f2 0xf296 0x905e 0xadd9 0x838d 0x1aed 0x0a30 0x0448
0x8950 0x6c10 0xe047 0x1d30 0x3c49 0x89d2 0x8086 0x2308
0x882f 0x7f18 0xc1ee 0x2e91 0x58be 0x2084 0x47a5 0x9c47
0x9f64 0x0356 0x01e8 0x0533 0xf1ac 0xe00d 0x106d 0x958e
0x880f 0x50ca 0x80d4 0xa322 0x85b2 0xe000 0x009c 0x389e
0x09c8 0x615a 0x199c 0x580e 0x7b3c 0xfaa5 0x39b4 0xb8cd
0x81a2 0xf2b2 0x6833 0x8116 0x70c4 0x6bb3 0x283f 0x9157
0x1225 0xeaa5 0x78e8 0x2b84 0x29ce 0xcb00 0x00db 0x40cd
0xa0fa 0xc0d4 0x4000 0x005d 0x401a 0x00d4 0xe0ce 0x4043
0xc0c3 0x80cf 0x40fd 0x80b1 0x606e 0x00aa 0xe0d1 0xc0ee
0x60e0 0x401f 0xa07c 0xe08f 0xe098 0x0000 0x0000